International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online)) Volume 7, Issue 5, May 2018)

Frequent Item Set using Apriori and Map Reduce algorithm: An Application in Inventory Management

Kranti Patil[1], Jayashree Fegade[2], Diksha Chiramade[3], Srujan Patil[4], Pradnya A. Vikhar[5]

[1,2,3,4,5]*KCES's COEIT, Jalgaon, Maharashtra, India.*

Abstract— **Data mining (DM) is a computerized technology that uses complicated algorithms to find relationships in large data bases Extensive growth of data gives the motivation to find meaningful patterns among the huge data. Sequential pattern provides us interesting relationships between different items in sequential database. Association Rules Mining (ARM) is a function of DM research domain and arise many researchers interest to design a high efficient algorithm to mine association rules from transaction database. It is a universal technique which uses to refine the mining techniques. In computer science and data mining, Apriori is a classic algorithm for learning association rules Apriori algorithm has been vital algorithm in association rule mining. Apriori algorithm is a realization of frequent pattern matching based on support and confidence measures produced excellent results in various fields. Main idea of this algorithm is to find useful patterns between different set of data. It is a simple algorithm yet having many drawbacks. So Apriori based MapReduce algorithm is proposed. Thus, there have been many approaches to convert many sequential algorithms to the corresponding Map/Reduce algorithms. Thus we presents Map/Reduce algorithm of the legacy Apriori algorithm that has been popular to collect the item sets frequently occurred in order to compose Association Rule in Data Mining. Theoretically, it shows that this algorithm provides high performance computing depending on the number of Map and Reduce nodes.**

The used Apriori based MapReduce algorithm will help in reducing multiple scans over the databases by cutting down unwanted transaction records for finding frequent itemsets.

Keywords—Map/Reduce, Apriori algorithm, Data Mining, Association Rule

I. Background

Data mining is the essential process of discovering hidden and interesting patterns from massive amount of data where data is stored in data warehouse, OLAP (on line analytical process), databases and other repositories of information. This data may reach to more than terabytes. Data mining is also called (KDD) knowledge discovery in databases, and it includes an integration of techniques from many disciplines such as statistics, neural networks, database technology, machine learning and information retrieval, etc. Interesting patterns are extracted at reasonable time by KDD's techniques. KDD process has several steps, which are performed to extract patterns to user, such as data cleaning, data selection, data transformation, data preprocessing, data mining and pattern evaluation.

Association rule are the statements that find the relationship between data in any database. Association rule has two parts "Antecedent" and "Consequent". Antecedent is that item which is found in the database, and consequent is the item that is found in combination with the first i.e. the antecedent. Association rule is used to abstract the data by picking the frequently used data in retail store for marketing, inventory control, etc.

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis.

Map/Reduce is an algorithm used in Artificial Intelligence as functional programming. It has been received the highlight since re-introduced by Google to solve the problems to analyze Big Data, defined as more than pita bytes of data in distributed computing environment. It is composed of two functions to specify, "Map" and "Reduce". They are both defined to process data structured in (key, value) pairs.

II. Introduction

Almost all industries stores huge quantities of their operational data in their local and distributed databases. Current developments are producing enormous amount of data day-by-day resulting in the need for persistent storage, analysis and efficient processing of the complex data. Data mining is a new powerful technology with great potential which discovers Information within the data that queries and reports can't effectively reveal. Current developments are producing enormous amount of data day-by-day resulting in the need for persistent storage, analysis and efficient processing of these complex data. Almost all industries stores huge quantities of their operational data in their local and distributed databases. These data can be used for analyzing customer trends which can be helpful in marketing the products to maximize profit and to efficiently manage the inventory.

The mining of Association Rules helps to extract interesting patterns, relationships or associations between the item sets in a transactional database. The Apriori Algorithm can be used for the mining of Association Rules which involves Frequent Item set Mining and Association Rule Generation over the operational data stored in transactional databases. However, processing of such voluminous data requires greater processing capabilities as the data is distributed in nature. Cloud computing helps in setting up an infrastructure required for distributed environment and enables companies to consume compute resources by increasing the capability of the shared resources. In this project we develop a programming model called MapReduce for processing massive datasets and it provides reliability, scalability and fault tolerance. In our work, we are implementing an efficient Apriori algorithm with MapReduce model.

III. Methodology

In this section we explain the overall view of the Apriori based MapReduce algorithm:

To implement an Apriori algorithm on MapReduce framework the main tasks are to design two independent map and reduce functions for the algorithm and to convert the datasets in the form of (key, value) pairs. In MapReduce programming, all the mapper and reducer on different machines execute in parallel fashion but the final result is obtained only after the completion of reducer. If algorithm is recursive, then we have to execute multiple phase of map-reduce to get the final result.

Phases of performing Apriori based MapReduce algorithm:

- **Map:**

Map is the name of a higher-order function that applies a given function to each element of a list.

- **Reduce:**

Reduce is the name of a higher-order function that analyze a recursive data structures and recombine through use of a given combining operation the results of recursively processing its constituent parts, building up a return value.

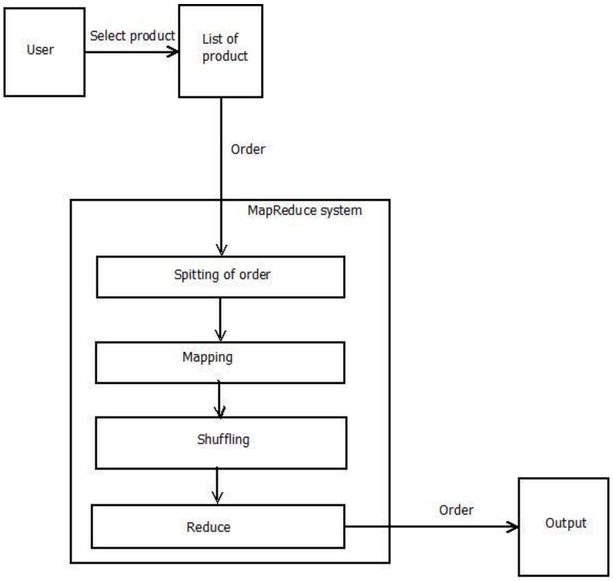


Figure1: Block Diagram for Apriori based MapReduce algorithm

Apriori algorithm is an iterative process and its two main components are candidate item sets generation and frequent item sets generation. In each scan of database, mapper generates local candidates and reducer sums up the local count and results frequent item sets. The count distribution parallel version of Apriori is best suited on MapReduce algorithm whereas to implement data distribution algorithm we have to control the distribution of data which is automatically controlled by MapReduce algorithm.

MapReduce takes an input, splits it into smaller parts, execute the code of the mapper on every part, then gives all the results to one or more reducers that merge all the results into one.

IV. IMPLEMENTATION

Here preprocessing means to be the preparation of datasets for identifying the missing, not applicable data values. Partitioning indicates the work of splitting up data to various data nodes and then the map and reduce functions are carried out.

The three important phases of reducer are:

Shuffle, Sort and Reduce.

During Mapper phase calculation, execution and distribution of data takes place. So this is very important to derive a strategy to deal with this issue. In this phase programmer writes his/her logic that will deal with the data. Mapper phase works parallel and to execute code as fast as possible.

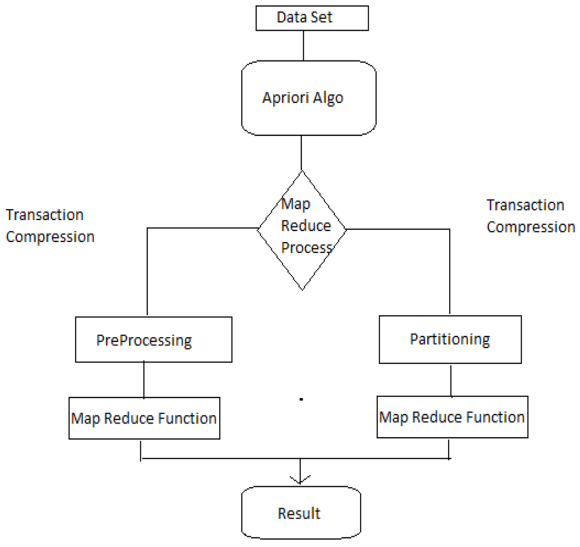


Figure 2: Architecture of Apriori and Mapreduce

A. MapReduce Algorithm

MapReduce is a parallel programming model designed for parallel processing of large volumes of data by breaking the job into independent tasks across a large number of machines. MapReduce is inspired form the list processing languages e.g. LISP. It uses two list processing idioms: map and reduce. Based on it a MapReduce program consists of two functions Mapper and Reducer which runs on all machines in a cluster. The input and output of these functions must be in form of (key, value) pairs.

1. The Mapper takes the input (k_1, v_1) pairs from HDFS and produces a list of intermediate (k_2, v_2) pairs. An optional Combiner function is applied to reduce communication cost of transferring intermediate outputs of mappers to reducers. Output pairs of mapper are locally sorted and grouped on same key and feed to the combiner to make local sum.

2. The intermediate output pairs of combiners are shuffled and exchanged between machines to group all the pairs with the same key to a single reducer. This is the only one communication step takes place and handle by the MapReduce platform. There is no other communication between mappers and reducers take place. The Reducer takes k_2, list (v_2) values as input, make sum of the values in list (v_2) and produce new pairs (k_3, v_3). Figure 2 illustrates the work flow of MapReduce.

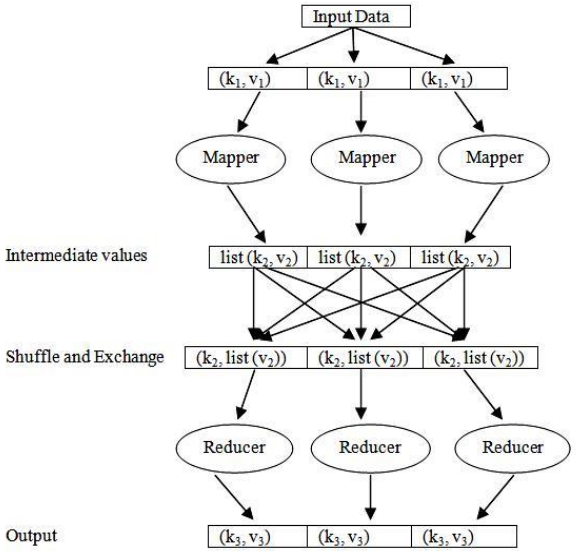


Figure 3: MapReduce Model.

MapReduce is a simplified programming model since all the parallelization, inter-machine communication and fault tolerance are handled by run-time system.

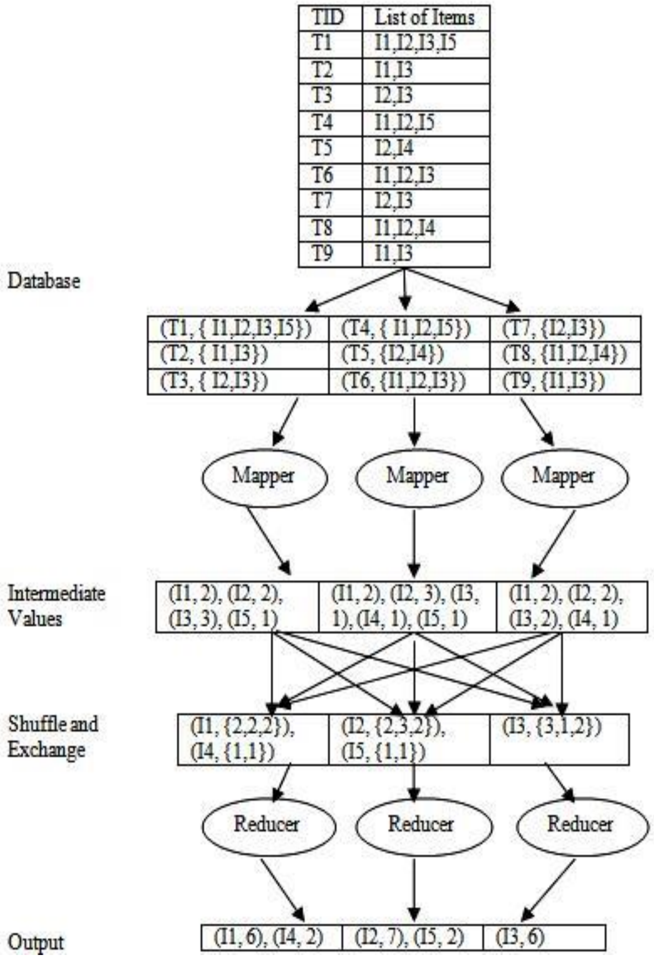
B. Apriori Algorithm on MapReduce

To implement an algorithm on MapReduce framework the main tasks are to design two independent map and reduce functions for the algorithm and to convert the datasets in the form of (key, value) pairs. In MapReduce programming, all the mapper and reducer on different machines execute in parallel fashion but the final result is obtained only after the completion of reducer. If algorithm is recursive, then we have to execute multiple phase of map-reduce to get the final result.

Apriori algorithm is an iterative process and its two main components are candidate item sets generation and frequent item sets generation. In each scan of database, mapper generates local candidates and reducer sums up the local count and results frequent item sets. The count distribution parallel version of Apriori is best suited on where as to implement data distribution algorithm we have to control the distribution of data which is automatically controlled by.

The first step of the algorithm is to generate frequent 1-itemsets L_1 which is illustrated in Figure 3 by an example. HDFS breaks the transactional database into blocks and distribute to all mappers running on machines. Each transaction is converted to (key, value) pairs where key is the TID and value is the list of items i.e. transaction. Mapper reads one transaction at time and output (key', value') pairs where key' is each item in transaction and value' i s 1. The combiner combines the pairs with same key' and makes the local sum of the values for each key. The output pairs of all combiners are shuffled & exchanged to make the list of values associated with same key, as (key, list (value)) pairs. Reducers take these pairs and sum up the values of respective keys. Reducers output (key, value) pairs where key' is item and value' is the support count ≥ minimum support, of that item. Final frequent 1-itemsets L_1 is obtained by merging the output of all reducers.

To generate frequent k-item sets L_k, each mapper reads frequent item sets L_{k-1} from previous iteration and generates candidate item sets C_k from L_{k-1} as in traditional algorithm. A candidate item set in C_k is selected as key and assigned a value 1, if it is present in the transaction assigned to the mapper. Now we have (key, value) pairs where key is k-item set and value is 1. All the remaining procedures are the same as generation of $L_{1.}$ Table 2 depicts the algorithms corresponding to mapper, combiner and reducer for Apriori algorithm.

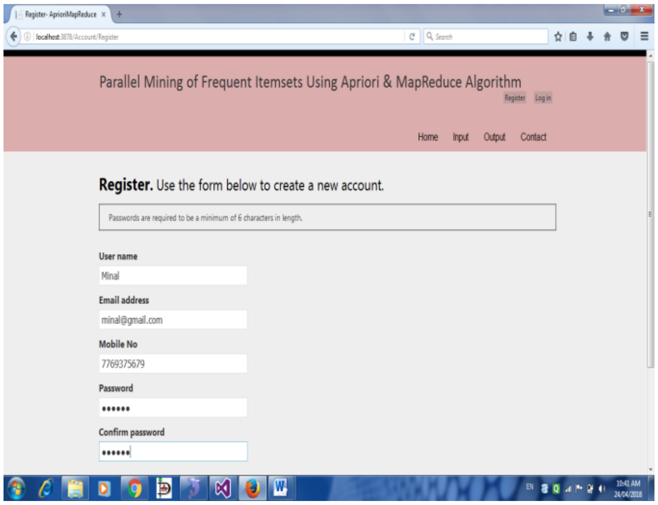


TID	List of Items
T1	I1,I2,I3,I5
T2	I1,I3
T3	I2,I3
T4	I1,I2,I5
T5	I2,I4
T6	I1,I2,I3
T7	I2,I3
T8	I1,I2,I4
T9	I1,I3

Figure 3: Generation of frequent 1 item set

V. RESULT

The Apriori based MapReduce algorithm is very useful for all industries. Almost all industries stores huge quantities of their operational data in their local and distributed databases. These data can be used for analyzing customer trends which can be helpful in marketing the products to maximize profit and to efficiently manage the inventory. Thus, Apriori based MapReduce algorithm can used which will help in reducing multiple scans over the databases by cutting down unwanted transaction records as well as redundant generation of sub – items while pruning the candidate item sets.



Parallel Mining of Frequent Itemsets Using Apriori & MapReduce Algorithm

Register. Use the form below to create a new account.

Passwords are required to be a minimum of 6 characters in length.

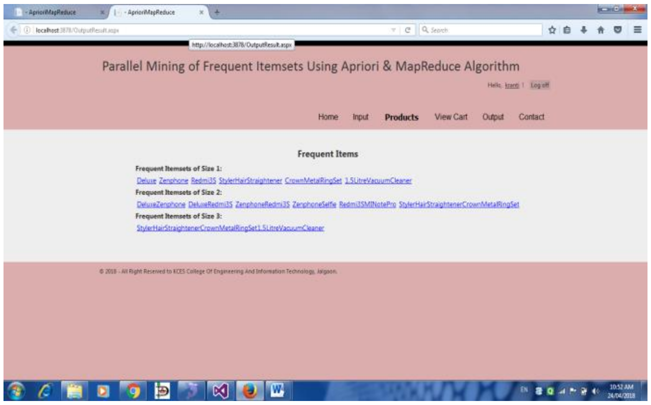
User name
Minal

Email address
minal@gmail.com

Mobile No
7769375679

Password
••••••

Confirm password
••••••



Parallel Mining of Frequent Itemsets Using Apriori & MapReduce Algorithm

Frequent Items

Frequent Itemsets of Size 1:
Deluxe Zenphone Redmi3S StylerHairStraightener CrownMetalRingSet 1.5LitreVacuumCleaner

Frequent Itemsets of Size 2:
DeluxeZenphone DeluxeRedmi3S ZenphoneRedmi3S ZenphoneSelfie Redmi3SMINotePro StylerHairStraightenerCrownMetalRingSet

Frequent Itemsets of Size 3:
StylerHairStraightenerCrownMetalRingSet1.5LitreVacuumCleaner

© 2018 - All Right Reserved to KCES College Of Engineering And Information Technology, Jalgaon.

VI. Conclusion

Data mining is emerged as an important area of development with the voluminous data generated during the transactions. The mining of Association Rules helps to extract interesting patterns, relationships or associations between the item sets in a transactional database. The Apriori Algorithm can be used for the mining of Association Rules which involves Frequent Item set Mining and Association Rule Generation over the operational data stored in transactional databases.

The project uses Apriori-Map/Reduce Algorithm which will help in reducing multiple scans over the databases by cutting down unwanted transaction records to gain higher performance than the sequential algorithm as the map and reduce nodes get added.

Same approach can be applicable in various areas like clinical big data analysis, Graph pattern matching, Geospatial Query processing for the identification of frequent item sets.

References

[1] S. Singh, R. Garg and P. K. Mishra, "Review of Apriori Based Algorithms on MapReduce Framework," *International Conference on Communication and Computing*, *Bangalore*, pp. 593–604, 2014.

[2] Jongwook Woo, “Apriori-Map/Reduce Algorithm”, Korean Technical Report of KISTI (Korea Institute of Science and Technical Information), Feb 2011

[3] Varsha Mashoria, Anju Singh, “Literature Survey on Various Frequent Pattern Mining Algorithm”, *IOSR Journal of Engineering (IOSRJEN)*, Vol. 3, PP 58-64, Jan. 2013.

[4] https://en.wikipedia.org/wiki/MapReduce.

[5] Jeffrey Dean and Sanjay Ghemawa, “MapReduce: Simplified Data Processing on Large Clusters", *Google Labs*, pp. 137–150, 2004.

[6] Ms. Pooja Agrawal, Mr. Suresh kashyap, Mr.Vikas Chandra Pandey, Mr. Suraj Prasad Keshri, “A Review Approach on various form of Apriori with Association Rule Mining”, *International Journal on Recent and Innovation Trends in Computing and Communication*, Volume: 1, pp. 462 – 468, May 2013

[7] Minal G. Ingle, N. Y. Suryavanshi “ Association Rule Mining using Improved Apriori Algorithm”, *International Journal of Computer Applications* Volume 112 – No 4, February 2015.

[8] R. Agrawal, R. Srikant et al., “Fast algorithms for mining association rules,” Proc. 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, pp. 487- 499, September 1994